Machine Learning

2022

Fiche pour l'UE Machine Learning, fait par ABITBOL Ethan et DUFOURMANTELLE Jeremy.

Sommaire

1 COURS 1

1) Tout d'abord, qu'est ce que le **machine learning ?**

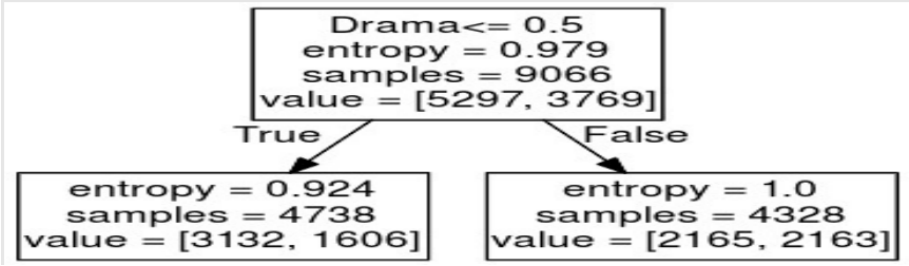
Le machine learning est une branche de l'intelligence artificielle englobant de nombreuses méthodes permettant de créer automatiquement des modèles à partir des données. Ces méthodes sont en fait des algorithmes.
On retrouve, principalement, deux categories du machine learning :

- **L'apprentissage Supervisé :** Les données sont deja étiquetées, le modèle de Machine Learning sait déjà ce qu'elle doit chercher (motif, élément...) dans ces données. À la fin de l'apprentissage, le modèle ainsi entraîné sera capable de retrouver les mêmes éléments sur des données non étiquetées. Parmi les algorithmes supervisés, on distingue les **algorithmes de classification** (prédictions non-numériques) et **les algorithmes de régression** (prédictions numérique) ou encore **les algorithmes de recommandations, ranking ou Forecasting**.
En fonction du problème à résoudre, on utilisera l'un de ces deux archétypes.

- **L'apprentissage NON Supervisé :** au contraire, consiste à entraîner le modèle sur des données sans étiquettes. La machine parcourt les données sans aucun indice, et tente d'y découvrir des motifs ou des tendances récurrents, on ne sais pas ce que l'on cherche.
Parmi les modèles non-supervisés, on distingue les **algorithmes de clustering** (pour trouver des groupes d'objets similaires), **d'association** (pour trouver des liens entre des objets) et **de réduction dimensionnelle** (pour choisir ou extraire des caractéristiques).

- Une troisième approche est celle de **l'apprentissage par renforcement.** Dans ce cas de figure, l'algorithme apprend en essayant encore et encore d'atteindre un objectif précis. Il pourra essayer toutes sortes de techniques pour y parvenir. Le modèle est récompensé s'il s'approche du but, ou pénalisé s'il échoue.

2) Qu'est ce qu'un **Arbre de décision ?**



- **Chaque noeud interne** : un **test** sur une dimension de X (- Est-ce que le film appartient au genre comédie ?)
- **Chaque branche** : un résultat du test.
- **Chaque feuille :** un label de Y

Classification en parcourant un chemin de la racine a une feuille.

3) Qu'est ce qu'une **Entropie :**

- Entropie d'une **Variable aleatoire:**
Soit X une variable aleatoire pouvant prendre n valeurs x_i:

$$H(X) = -\sum_{i=1}^{n} P(X = x_i) * log[P(X = x_i)]$$

Plus l'entropie est grande, plus le desordre est grand.
Entropie nulle $\rightarrow$ pas d'alea.

- Entropie d'**un échantillon : cas binaire**:
X ensemble de donnée avec Y leur etiquette (positif / négatif),
p_+ la proportion d'exemples positifs,
p_- la proportion d'exemples négatifs,

$$H(Y) = -p_+ * log(p_+) - p_- * log(p_-)$$

- Entropie **conditionnelle**:
Entropie conditionnelle : $H(Y|X) = \sum_{i=1} P(X = x_i) * H(Y|X = x_i)$
Dans notre cas, en faisant un test T sur un des attributs, on obtient deux partitions d'exemples de X : X1 qui verifie le test et X2 qui ne verifie pas le test (resp. Y1 et Y2). L'entropie conditionnelle au test T est :

$$H(Y|T) = \frac{|X1|}{|X|} * H(Y1) + \frac{|X2|}{|X|} H(Y2)$$

2 TD1

2.1 Exercice 1 - Echauffement : probas discretes, continues

2.1.1 Q1.1.1

- Un **événement élémentaire** est le résultat d'une experience et est une partition d'un événement,
- Un **événement** est une partition de l'univers, il peut etre composé de plusieurs événement élémentaires,
- Un **univers** est composé d'événements élémentaires

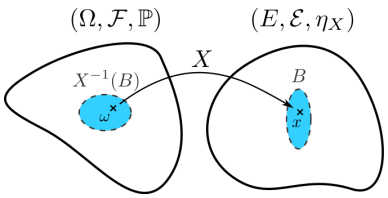
2.1.2 Q1.1.2

Soit Ω, l'ensemble des événements de l'univers. Pour construire un espace probabilisé sur un ensemble E dénombrable, on définit une fonction P qui prends un élément de Ω et qui renvoit un réel dans $[0, 1]$.
Tel que : $\forall x \in \Omega \; P : P(x) \rightarrow [0, 1]$.
Cette fonction de probabilité doit respecter les axiomes suivants :

1) $P(\Omega) = 1 \rightarrow P(\oslash) = 0$
2) $\forall x \in \Omega \; P(x) \geq 0$
3) si $A \cap B = \oslash$ (incompatibilité) alors $P(A \cup B) = P(A) + P(B)$

2.1.3 Q1.1.3

Une variable aléatoire est une "passerelle" entre l'ensemble de tous les événements Ω et un ensemble dans $\mathbb{R}$. Alors on a X une variable aléatoire définit tel que $X : \Omega \rightarrow \mathbb{R}$. C'est grâce à cette "passerelle" que l'on peut parler d'esperance, de variance, de moyenne etc...



$X(w) = x$
$P(X = x) = P(X^{-1}(x))$
$P(X \in [a, b]) = P(X^{-1}([a, b]))$

2.1.4 Q1.2

La loi qui permet de modéliser la variable aléatoire indiquant une série d'echecs puis avoir un succès est la loi Géométrique. On peut définir une variable aléatoire suivant une loi Géométrique $G(p)$ ou p est la probabilité du succès tel que : $X \sim G(p)$. On définit alors sa variance et son esperance :

$$P(X = k) = (1-p)^{k-1} * p \text{ avec k le nombres d'épreuves}$$
$$E(X) = \frac{1}{p}$$
$$V(X) = \frac{1-p}{p^2}$$

2.1.5 Q1.3

Soit un jeu de 52 cartes, on définit les événements suivants :

A : Tirer un roi
B : Tirer un pique
$A \cap B$: Tirer un roi de pique
$A \cup B$: Tirer un roi ou un pique

On a alors les valeurs de probabilités suivantes :

$$P(A) = \frac{4}{52} = \frac{1}{13}$$
$$P(B) = \frac{13}{52} = \frac{1}{4}$$
$$P(A \cap B) = \frac{1}{52}$$
$$P(A \cup B) = \frac{3}{52} + \frac{12}{52} + \frac{1}{52} = \frac{16}{52}$$

On souhaite savoir si les événements A et B sont indépendants, pour cela il faut pouvoir vérifier l'égalité : $P(A \cap B) = P(A)P(B)$

$$\frac{1}{52} = \frac{1}{13} * \frac{1}{4}$$

Donc les événements A et B sont bien indépendants et il sont bien compatible car $P(A \cap B) \neq \oslash$

2.2 Exercice 2 - Probabilités continues

On considère dans cette exercice les variables aléatoire X et Y représentant respectivement le niveau d'embouteillage et la météo à un instant donné.
L'univers est donc continu. Les événements sont infinis et non dénombrables,$P([a, b[)$ avec [a,b[un intervalle et non un sous ensemble de Ω.

2.2.1 Q2.1

Une densité de probabilité représente comment la probabilité évolue au voisinage du point dont l'on souhaite connaitre la probabilité. On la définit de la manière suivante :

$$P(X \in [a, b[) = P_X([a, b[) = \int_a^b p_X(x)dx$$

Nous avons donc la propriété suivante :

$$P(X \in \mathbb{R}) = \int_{\mathbb{R}} p_X(x)dx = 1$$

On peut exprimer la densité jointe de deux lois X et Y de la façon suivante :

$$P((X, Y) \in I) = \int_I p_{X,Y}(x, y)dxdy$$

On peut exprimer la densité marginale de deux lois X et Y de la façon suivante :

$$P(X \in I) = \int_I p_{X,Y}(x, y)dy$$

2.2.2 Q2.2

Soit X une variable aléatoire sur $\mathbb{R}$ tel que $X : \Omega \to \mathbb{R}$ de l'espace probabilisé $(\Omega, \epsilon, \mathbb{P})$. On définit l'esperance de X par une intégrale sur Ω de la maniere suivante :

$$E(X) = \int_{\mathbb{R}} x p_X(x) dx,$$

On définit l'esperance de X par une intégrale sur Ω de la maniere suivante :

$$E(X) = \int_{\Omega} X(w) p_X(w) dw,$$

2.2.3 Q2.3

On exprime $E(X|Y = y)$:

$$E(X|Y = y) = \int_{\mathbb{R}} x p_X(x|Y = y) dx$$

2.2.4 Q2.4

Rappel sur la loi des grands nombres :

$$\forall \epsilon > 0 \ \lim_{n \to +\infty} P(|\frac{1}{n} \sum_{i=1}^{n} x_i - E(X)| > \epsilon) = 0$$

Donc, $E(X)$ tend vers $\frac{1}{|E|} \sum_i x_i$ d'après la loi des grands nombres.

2.3 Exercice 3 - Classifieur Bayesien

2.3.1 Q3.1.1

Un vecteur de vote contient 16 votes. Chaque vote peut valoir 3 valeurs différentes (Non,Oui,NSP). On a donc 3^{16} votes différents possible.

2.3.2 Q3.1.2

Soit un vecteur de vote V, pour estimer si le représentant R est démocrate ou républicain, on cherche à calculer le rapport T:

$$T = \frac{P(R=Démocrate|Vote=V)}{P(R=Républicain|Vote=V)}$$

Si $T > 1$ alors le vote V est celui d'un représentant démocrate sinon c'est celui d'un républicain.

Nous ne pouvons pas calculer cette probabilité directement car nos deux tableaux représente les distributions de $P(Vote|R = Démocrate)$ et $P(Vote|R = Républicain)$. Il nous manque donc des données, pour ce faire nous allons passer par le théorème de Bayes :

$$P(R|V) = \frac{P(V|R)P(R)}{P(V)}$$

Ce qui nous donne :

$$\frac{P(R=Démocrate|Vote=V)}{P(R=Républicain|Vote=V)} = \frac{\frac{P(Vote=V|R=Démocrate)P(R=Démocrate)}{P(Vote=V)}}{\frac{P(Vote=V|R=Républicain)P(R=Républicain)}{P(Vote=V)}} = \frac{P(Vote=V|R=Démocrate)P(R=Démocrate)}{P(Vote=V|R=Républicain)P(R=Républicain)}$$

Nous connaissons la distribution de $P(R)$:

$$P(R = Républicain) = \frac{|Républicain|}{|Républicain|+|Démocrate|} \ P(R = Démocrate) = \frac{|Démocrate|}{|Républicain|+|Démocrate|}$$

Il nous reste plus qu'à calculer la distribution $P(Vote = V|R)$. Cette distribution ne peut pas être estimée, nous faisons donc une hypothèse d'indépendance entre les votes de V. On pose donc :

$$P(Vote = V|R) = \prod_{i=1}^{16} P(v_i|R)$$

Les valeurs de probabilités étant très faible, nous passons les quantités calculées au log, ce qui ne change en rien la solution. On pose donc :

$$log(\frac{P(Vote=V|R=Démocrate)P(R=Démocrate)}{P(Vote=V|R=Républicain)P(R=Républicain)}) = log(\frac{[\prod_{i=1}^{16} P(v_i|R=Démocrate)]*\frac{|Démocrate|}{|Républicain|+|Démocrate|}}{[\prod_{i=1}^{16} P(v_i|R=Républicain)]*\frac{|Républicain|}{|Républicain|+|Démocrate|}})$$
$$= log(\frac{[\prod_{i=1}^{16} P(v_i|R=Démocrate)]*|Démocrate|}{[\prod_{i=1}^{16} P(v_i|R=Républicain)]*|Républicain|})$$
$$= [\sum_{i=1}^{16} log(P(v_i|R = Démocrate)] - [\sum_{i=1}^{16} log(P(v_i|R = Répubicain)] + log(\frac{|Démocrate|}{|Républicain|})$$

2.3.3 Q3.2.1

Soit G la variable aléatoire indiquant le genre d'une personne. G peut donc valoir Homme ou Femme. Dans ce cas G suit une loi de bernouilli de paramètre p.
Ici p désigne la probabilité d'être un homme donc : $p = \frac{|h|}{|h|+|f|}$

2.3.4 Q3.2.2

On nomme T la variable aléatoire donnant la taille d'une personne choisi au hasard. D'après la formule des probabilités totale, on pose :

$$P(T = t) = P(T = t|G = Homme)P(G = Homme) + P(T = t|G = Femme)P(G = Femme)$$

De plus d'après l'énoncé, nous savons que la répartition des tailles sont gaussiennes au sein de chaque sous-population, donc chez les Hommes et chez les Femmes. Ce qui nous donne les distributions suivantes :

$$P(T = t|G = x) \sim N(\mu_x, \sigma_x^2) \text{ ou } x \in \{Femme, Homme\}$$

Donc on obtient :

$$P(T = t|G = x) = \frac{1}{\sigma_x\sqrt{2\pi}} \exp - \frac{(t-\mu_x)^2}{2\sigma_x^2}$$

En remplacant :

$$P(T = t) = \frac{1}{\sigma_{Homme}\sqrt{2\pi}} \exp - \frac{(t-\mu_{Homme})^2}{2\sigma_{homme}^2} * p + \frac{1}{\sigma_{Femme}\sqrt{2\pi}} \exp - \frac{(t-\mu_{Femme})^2}{2\sigma_{femme}^2} * (1 - p)$$

On souhaite calculer la distribution $P(G = Femme|T = t)$, d'après le thérorème de Bayes :

$$P(G = Femme|T = t) = \frac{P(T=t|G=Femme)P(G=Femme)}{P(T=t)}$$

2.3.5 Q3.2.3

Classifieur Bayesien optimal pour classifier un Homme ou une Femme en fonction d'une taille t :
On applique le rapport :

$$Rapport = \frac{P(G=Femme|T=t)}{P(G=Homme|T=t)} = \frac{P(T=t|G=Homme)P(G=Homme)}{P(T=t|G=Femme)P(G=Femme)}$$

On procède au calcul comme à la question **Q3.1.2**. Si $Rapport > 1$ alors le genre est Féminin sinon le genre est Masculin.

2.3.6 Q3.2.4

TODO

2.3.7 Q3.3

Il y a en tout 5 paramètres pour le classifieur : 2 pour chaque distribution gaussienne $(\mu_{Homme}, \mu_{Femme}, \sigma_{Homme}, \sigma_{Femme})$ et 1 paramètre pour la prior p

2.4 Exercice 4 - Entropie

2.4.1 Q4.1.1

- n_k : nombre de fois où k apparait dans le tirage.
- p_k : probabilité de tirer le numéro k.

Probabilité de tirer la suite $(x_1, x_2, ..., x_m)$:

$$P((x_1, x_2, ..., x_m)) = \prod_{i=1}^{6} p_i^{n_i}$$

2.4.2 Q4.1.2

D'après la loi des grands nombre (binomiale) :

$$\lim_{n\to+\infty} n_k = np_k$$

Donc en déduit :

$$\lim_{n\to+\infty} \prod_{i=1}^{6} p_i^{n_i} = \prod_{i=1}^{6} p_i^{np_i}$$

$$= \prod_{i=1}^{6} 2^{n*p_i log(p_i)}$$

$$= 2^{n \sum_{i=1}^{6} p_i log(p_i)}$$

$$= 2^{-nH(p)}$$

Nous remarquons qu'une entropie très forte siginifie que la séquence de tirage va être très diversifié (e.g semblable à une distribution uniforme entre les tirages). Au contraire, une entropie faible, proche de 0 signifie que le tirage va contenir peu de diversification. Exemple : une distribution de deux classes $1, 2$ où 1 représente 80% de l'effectif va donner une entropie proche de 0. Cependant, si la distribution est $50/50$ l'entropie va être beaucoup plus elevée.

2.4.3 Q4.2.1

Montrons que $H(p) \geq 0$:

$$\text{Comme } p_i \in]0; 1], log(p_i) \leq 0$$

$$-log(p_i) \geq 0$$

$$H(p) = -\sum_{i=1}^{n} p_i * log(p_i) \geq 0$$

2.4.4 Q4.2.2

Soient **p** et **q** deux vecteurs de probabilité de même dimension. Montrons que $-\sum_{i=1}^{n} p_i * log(p_i) \leq -\sum_{i=1}^{n} p_i * log(q_i)$.
On pose :

$$-\sum_{i=1}^{n} p_i * log(p_i) \leq -\sum_{i=1}^{n} p_i * log(q_i)$$

$$\sum_{i=1}^{n} p_i * log(p_i) - \sum_{i=1}^{n} p_i * log(q_i) \geq 0$$

$$-(\sum_{i=1}^{n} p_i * log(p_i) - \sum_{i=1}^{n} p_i * log(q_i)) \leq 0$$

$$\sum_{i=1}^{n} p_i * log(q_i) - p_i * log(p_i) \leq 0$$

$$\sum_{i=1}^{n} p_i * (log(q_i) - log(p_i)) \leq 0$$

$$\sum_{i=1}^{n} p_i * log(\frac{q_i}{p_i}) \leq 0$$

Prouvons que cette derniere inéquation est vérifiée;

Pour cela, trouvons un majorant de la fonction $log(x)$:

Nous savons que la fonction $log(x)$ est concave, donc $\forall a, T(x) = f'(a)(x-a) + f(a) \geq log(x)$ où la fonction $T(x)$ est l'équation de la tangente de la fonction $log(x)$ au point d'abscisse a.

Nous calculons alors l'équation de la tangente au point d'abscisse 1, car nous savons que $log(1) = 0$ et cela nous permettra de simplifier les calculs.

$$f(x) = log(x)$$

$$f'(x) = \frac{1}{x}$$

Donc,

$$T(x) = \frac{1}{1} * (x-1) + log(1) = x - 1$$

On obtient l'inéquation :

$$x - 1 \geq log(x)$$

On remplace :

$$\sum_{i=1}^{n} p_i * log(\frac{q_i}{p_i}) \leq \sum_{i=1}^{n} p_i * (\frac{q_i}{p_i} - 1)$$

$$\sum_{i=1}^{n} p_i * log(\frac{q_i}{p_i}) \leq \sum_{i=1}^{n} q_i - \sum_{i=1}^{n} p_i$$

Par hypothèse sur les vecteurs **p** et **q**, nous savons que la somme de leurs coefficients est égale à 1. En conséquence, on prouve que l'inégalité est vérifié car,

$$\sum_{i=1}^{n} q_i - \sum_{i=1}^{n} p_i = 0$$

2.4.5 Q4.2.3

On sait que :

$$H(p) = -\sum_{i=1}^{n} p_i * log(p_i)$$

D'après la question **Q4.2.2**,

$$H(p) = -\sum_{i=1}^{n} p_i * log(p_i) \leq -\sum_{i=1}^{n} p_i * log(q_i).$$

On considère que les q_i valent $\frac{1}{n}$ pour faire nos calculs, car l'inégalité est vérifié pour n'importe quelle valeur de q_i.

On pose donc,

$$H(p) \leq -\sum_{i=1}^{n} p_i * log(\frac{1}{n})$$

$$H(p) \leq -\sum_{i=1}^{n} p_i * (log(1) - log(n))$$

$$H(p) \leq \sum_{i=1}^{n} p_i * log(n)$$

Comme on sait par hypothèse que $\sum_{i=1}^{n} p_i = 1$, on obtient:

$$H(p) \leq log(n)$$

2.4.6 Q4.3.1

TODO

2.4.7 Q4.3.2

TODO

2.4.8 Q4.3.3

TODO

2.4.9 Q4.3.4

TODO

3 TD2

3.1 Exercice 1 - Classifieur bayésien

Soit $\mathbb{X} = \{x_1, ..., x_l\}$, $\forall x_i \in \mathbb{X}, x_i \in \mathbb{R}^d$ un ensemble de description dans $\mathbb{R}^d$ et $\mathbb{Y}$ l'ensemble des labels, tel que $\mathbb{Y} = \{y_1, ..., y_l\}$

3.1.1 Q1.1

Un classifieur est une fonction qui pour un exemble $x_i \in \mathbb{X}$ associe une classe/label. Le classifieur bayésien cherche à trouver la meilleur classe d'un exemple en se basant sur un critère bayesien comme la vraisemblance **à posteriori**, tel que :

$$\hat{y} = argmax_y P(y|x)$$

On définit aussi les notations d'une classe prédite: $\hat{y} = y_{pred} = f(x)$

Dans un classifieur, nous donnons les noms suivants aux probabilités :

$P(y)$: **La prior**

$P(x)$: **L'évidence**

$P(x|y)$: **La vraisemblance à priori**

$P(y|x)$: **La vraisemblance à posteriori**

3.1.2 Q1.2

Estimation de l'erreur faite par le classifieur bayésien f à un point x :

$$\text{Erreur}(f,x) = P(f(x) \neq y|x)$$

$$\text{Erreur}(f,x) = 1 - P(f(x) = y|x)$$

$$\text{Erreur}(f,x) = 1 - max_y P(y|x)$$

3.1.3 Q1.3

On définit le cout $\lambda(y_{pred}, y)$ d'une erreur (aussi appelé fonction de perte) consistant à prédire le label y_{pred} plutot que y. Dans le cas de l'erreur 0-1, c'est à dire que si y_{pred} et y sont différents alors on a un cout de 1 et 0 sinon. Nous définissons alors fonction :

$$\lambda(y_{pred}, y) = \begin{cases} 1 & \text{si } y_{pred} \neq y \\ 0 & \text{sinon.} \end{cases}$$

Un cout asymétrique est une fonction qui renvoit différentes valeurs pour différentes classes qui ne sont pas pareilles. On peut prendre par exemple dans un cadre de classification multiclasse le cout entre un chat et un chien qui sera plus petit que un cout entre un chat et un camion.

$$\lambda(chat, chien) < \lambda(chat, camion)$$

Dans le cadre de classification binaire, on peut prendre l'exemple de cout entre la prédiction de faux-positif et faux-négatif. En effet la valeur du cout d'un faux négatif(-) sera beaucoup plus grande que le cout d'un faux-positif(+).

$$\lambda(+, -) < \lambda(-, +)$$

3.1.4 Q1.4

Nous définissons le risque $R(y_i|x)$ de prédire une classe y_i.

$$R(y_i|x) = \sum_j \lambda(y_j, y_i)P(y_j|x)$$

Dans le cadre d'une classification binaire (deux classes y_0, y_1), nous obtenons par exemple:

$$R(y_0|x) = \lambda(y_0, y_0)P(y_0|x) + \lambda(y_1, y_0)P(y_1|x) = P(y_1|x) = 1 - P(y_0|x)$$

$$R(y_1|x) = \lambda(y_0, y_1)P(y_0|x) + \lambda(y_1, y_1)P(y_1|x) = P(y_0|x) = 1 - P(y_1|x)$$

De cette facon, nous en déduisons :

$$R(y_i|x) = 1 - P(y_i|x)$$

3.1.5 Q1.5

Nous définissons le risque $R(f)$ associé au classifieur f.

$$R(f) = \int_x R(f(x)|x)p(x)dx$$

3.1.6 Q1.6

Voici différent critére de décision pour le classifieur f.

En fonction du cout λ et de la vraissemblance à posteriori :

$$f(x) = argmin_y R(y|x)$$

$$f(x) = argmin_y \sum_i \lambda(y_i, y)P(y_i|x)$$

En fonction du cout λ uniquement :

$$f(x) = argmin_y\{\lambda(y_i, y) : y \in Y\}$$

En fonction de la prior $P(y)$:

$$f(x) = argmax_y P(y)$$

En fonction du maximum de vraisemblance a posteriori (qui est le critère de classification du classifieur bayésien):

$$f(x) = argmax_y P(y|x)$$

3.2 Exercice 2 - Estimation de densité

3.2.1 Q2.1

L'estimation de la densité p_b d'une variable aléatoire X est :

$$p(X \in B) = \int_{x \in B} p(x)dx$$

Nous faisons ensuite l'hypothèse que si l'espace B est suffisamment petit alors $p(x) = p_B$. On obtient donc :

$$p(X \in B) = \int_{x \in B} p_B dx = p_B V$$

On cherche a connaître la densité de l'espace B, donc on cherche à calculer :

$$\mathbb{E}(\sum_{i=1}^{n} X_i) = nP(X \in B)$$

Finalement, on obtient : $P_b = \frac{k}{nV}$
avec k : nombres d'échantillons observées dans la zone B
n : parmis n échantillons tirés
V : volume de la zone B

3.2.2 Q2.2

La méthode des histogrammes:
Pour chaque rectangle r_{ij}, on estime :

$$P_{ij} = \frac{k_{ij}}{nV} = \frac{k_{ij}}{n\Delta_x\Delta_y}$$
$$\text{avec } \Delta_x = \frac{x_{max}-x_{min}}{n_x} \text{ et } \Delta_y = \frac{y_{max}-y_{min}}{n_y}$$

n_i : nombre de bins sur l'axe i.
k_{ij} : nombre d'élément dans la case (i, j).
n : nombre d'élément total

3.2.3 Q2.3

Méthodes d'estimation de densité à noyaux :
Pour introduire ces méthodes, présentons l'estimation est faite en centrant une fenêtre autour du point d'intérêt (l'idée des fenêtre de parzen, dans un espace de dimension d, ce qui nous donne un hypercube).
La densité en un point x_0 est défini par :

$$p(x_0) = \frac{\frac{k}{N}}{r^d}$$

k : Nombre de points dans l'hypercube centré en x_0
N : Nombre total d'élément dans l'échantillon
r : Longueur de l'hypercube, donc r^d est le volume V.

Sauf que en réalité on ne connait pas k à l'avance (le nombre de points dans le voisinnage de x_0).

Pour cela, nous allons grâce à une fonction indicatrice calculer pour tous les points, si ils sont dans l'hypercube ou non.

$$\phi(x) = \begin{cases} 1 & \text{si } |x_i| \leq 1/2 \\ 0 & \text{sinon.} \end{cases}$$

Donc un point x est dans l'hypercube centré en x_0 si et seulement si $\phi(\frac{x_0-x}{r}) = 1$.

Nous pouvons maintenant calculer ce k avec tous les points de l'échantillon, tel que :

$$k = \sum_{i=1}^{N} \phi(\frac{x_0-x_i}{r})$$

En remplacant dans la formule, on obtient donc :

$$p(x_0) = \frac{1}{NV} \sum_{i=1}^{N} \phi(\frac{x_0-x_i}{r})$$

L'idée des méthodes à noyaux est de remplacer la fonction indicatrice $\phi(x)$ (hypercube) par une fonction de densité centrée en x_0 telle que une fonction gaussienne ou uniforme par exemple.

3.3 Exercice 3 - Classification selon voisinage

3.3.1 Q3.1

Fenêtres de Parzen : on choisis la classe majoritaire de l'hypercube centré en x.
K-nn: On choisis la classe majoritaire parmis les K plus proches voisins.
Quand on tend le nombre d'échantillon vers l'infini, on tend vers un classifieur Bayésien

3.3.2 Q3.2

La frontiere va se situer à la distance moyenne des points opposés car un point à classifier aura à choisir uniquement 1 voisin le plus proche.

3.3.3 Q3.3

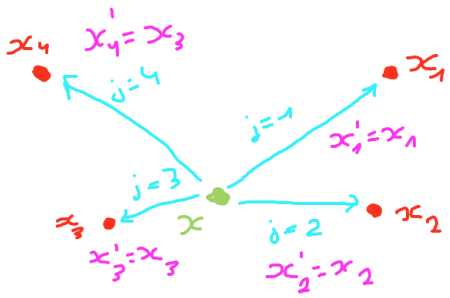
Si l'outlier est une croix, alors la frontiere de décision va s'agrandir pour la zone des croix en la prenant en considération et donc l'englober. Meme chose si l'outlier est un triangle.

3.3.4 Q3.4

Si $k = 3$, alors la frontiere va évoluer de sorte a prendre en compte 3 voisins plus proches est devenir plus descriptive avec plus de discontinuité.
Quand k tend vers l'infini, nous devons alors prendre la classe majoritaire. A ce moment, il n'y a plus de frontiere car la réponse du classifieur sera toujours la même.

3.3.5 Q3.5

Montrons que la séquence $\{x'_i\}_{i=1}^n$ converge vers l'exemple à classifier x. Pour cela, illustrons la convergence de cette séquence par un schéma :



Notre échantillon aléatoire $\{x_1, ..., x_n\}$ est en rouge, les étapes j de l'algorithme sont en bleus et le plus proche voisin de x à chaque étape j est en violet.

L'algorithme va chercher à chaque itération à choisir entre le plus proche voisin déja rencontré et le nouvel exemple de l'échantillon x_i, tel que $x'_j = min(x'_{j-1}, x_i)$. Nous savons aussi que $min(x'_j, x) = x$. Nous en déduisons donc que $\lim_{i \to +\infty} x'_i = x$.

3.3.6 Q3.6

D'après l'énoncé, nous pouvons définir le risque comme :

$$R(x, x'_n) = P(y \neq y'_n | x, x'_n)$$

Ou y'_n $(= f(x'_n))$ représente la classe de l'exemple x'_n.
Ce qui nous donne,

$$R(x, x'_n) = 1 - P(y = y'_n | x, x'_n)$$

L'égalité $y = y'_n$ est vérifié a chaque fois que y et y'_n sont de la même classe, nous simplifions donc par ;

$$= 1 - \sum_k P(y = k \wedge y'_n = k | x, x'_n)$$

Les probabilités étant indépendantes, on simplifie,

$$= 1 - \sum_k P(y = k|x) P(y'_n = k|x'_n)$$
$$= 1 - \sum_k q_k(x) q_k(x'_n)$$

3.3.7 Q3.7

D'après la question **3.5**, nous savons que $\lim_{i \to +\infty} x'_i = x$. Donc, on en déduit :

$$\lim_{n \to +\infty} P(y'_n = k | x'_n) = P(y = k | x)$$

On obtient alors,

$$r(x) = \lim_{n \to +\infty} R(x, x'_n)$$

$$= 1 - \sum_k P(y = k|x) P(y = k|x)$$

$$= 1 - \sum_k q_k(x)^2$$

3.3.8 Q3.8

TODO

3.3.9 Q3.9

TODO

4 TD 3 : Descente de gradient, Modèles linéaires

4.1 Exercice 1 – Apéro

4.1.1 Q1.1

Qu'est ce qu'une fonction convexe ?
Il suffit de calculer la dérivé seconde et de montrer qu'elle est positive :

$$f'' > 0$$

- $f(x) = x * cos(x)$ s'annule plusieurs fois donc n'est pas connexe.
Nb: Si un fonction s'annule plus de 2 fois, elle n'est pas convexe.

- $g(x) = -log(x) + x^2$ comme $log(x)$ est une fonction convexe et que x^2 l'est aussi alors $g(x)$ est convexe.
Nb: L'addition de deux fonctions convexe est convexe.

- $h(x) = x\sqrt{x} = (3)^{\frac{3}{2}}$ et $h''(x) = \frac{3}{4} * x^{-\frac{1}{2}} > 0$ donc $h(x)$ est convexe.
Nb: Le produit de deux fonctions convexe n'est pas forcement convexe.

- $t(x) = -log(x) - log(10 - x)$ et $t'' = \frac{1}{x^2} + \frac{1}{(10-x)^2 > 0}$ donc $t(x)$ est convexe.
Nb: Une composition de fonction convexe n'est pas forcement convexe.

4.1.2 Q1.2

Le gradient de f : $\nabla f(x)$ est :

$$\nabla_x f(x) = \begin{pmatrix} df/dx_1 \\ . \\ . \\ . \\ df/dx_d \end{pmatrix} = \begin{pmatrix} 2 \\ 2x_2 + x_3 \\ x_2 \end{pmatrix}$$

4.1.3 Q1.3

$$\nabla_x(f(x) + g(x)) = \begin{pmatrix} d(f+g)/dx_1 \\ . \\ . \\ . \\ d(f+g)/dx_d \end{pmatrix} = \begin{pmatrix} \frac{df}{dx_1} + \frac{dg}{dx_1} \\ . \\ . \\ . \\ \frac{df}{dx_d} + \frac{dg}{dx_d} \end{pmatrix} = \nabla_x f(x) + \nabla_x g(x)$$

Avec $t \in \mathbb{R}$:

$$\nabla_x t f(x) = \begin{pmatrix} dt(f)/dx_1 \\ . \\ . \\ . \\ dt(f)/dx_d \end{pmatrix} = \begin{pmatrix} td(f)/dx_1 \\ . \\ . \\ . \\ td(f)/dx_d \end{pmatrix} = t\nabla_x f(x)$$

Avec $b \in \mathbb{R}^d$:

$$\nabla_x b^T x = \nabla_x(b_1x_1 + ... + b_dx_d) = \begin{pmatrix} d(b_1x_1 + ... + b_dx_d)/dx_1 \\ . \\ . \\ . \\ d(b_1x_1 + ... + b_dx_d)/dx_d \end{pmatrix} = \begin{pmatrix} b_1 \\ . \\ . \\ . \\ b_d \end{pmatrix} = b$$

Avec A une matrice symetrique :

$$\nabla_x x^T A x = \nabla_x \begin{pmatrix} x_1 & ... & x_d \end{pmatrix} \begin{pmatrix} \sum_{j=1}^{d} a_{1,j} x_j \\ . \\ . \\ . \\ \sum_{j=1}^{d} a_{d,j} x_j \end{pmatrix} = \nabla_x \sum_{i=1}^{d} x_i * (\sum_{j=1}^{d} a_{i,j} x_j) = 2Ax$$

4.2 Exercice 2 – Régression linéaire

4.2.1 Q2.1.1

Définition : un modèle de régression linéaire est un modèle de régression qui cherche à établir une relation linéaire entre une variable, dite expliquée, et une ou plusieurs variables, dites explicatives. Expliquer une variable Y en fonction d'une variable X (resp. plusieurs variables $X_1, X_2, ..., X_q$), cette méthode est utilisé dans le but de la prédiction.

On rappelle l'écriture de la fonction linéaire f paramétré par w :

$$f_w(x) = w_1 x_{i,1} + ... + w_d x_{i,d} \text{ avec w le poids}$$
$$f_w(X) = X.W$$

La fonction d'erreur qui est utilisé est l'erreur des moindres carré.

Sous la forme indicielle :

$$MSE(W) = \sum_{i=1}^{n} (f_w(x_i) - y_i)^2 = \sum_{i=1}^{n} ((w_1 x_{i,1} + ... + w_d x_{i,d}) - y_i)^2$$

4.2.2 Q2.1.2

Voici les dimensions des matrices utilisées :
- $X \in \mathbb{R}^{n,d}$
- $Y \in \mathbb{R}^n$
- $W \in \mathbb{R}^d$

Erreur des moindres carrés sous forme matricielle :

$$MSE(W, X, Y) = (X.W - Y)^T (X.W - Y) = ||XW - Y||^2 = \begin{pmatrix} f_w(x_1) - y_1 & ... & f_w(x_n) - y_n \end{pmatrix} \begin{pmatrix} f_w(x_1) - y_1 \\ . \\ . \\ . \\ f_w(x_n) - y_n \end{pmatrix}$$

4.2.3 Q2.1.3

Nous cherchons à trouver w^* qui minimise l'erreur MSE de notre fonction f. Pour cela nous allons calculer le gradient de la fonction de cout MSE, puis l'annuler pour obtenir w^*.

$$\nabla_w MSE(W) = \nabla_w (X.W - Y)^T (X.W - Y)$$

Distribution de la transposée

$$= \nabla_w ((X.W)^T - Y^T)(X.W - Y)$$

Distribution de la partie de gauche

$$= \nabla_w (X.W)^T (X.W) - (X.W)^T.Y - Y^T (X.W) + Y^T.Y$$

Nous savons que $\nabla_w Y^T.Y = 0$ car ne dépend pas de w, donc

$$= \nabla_w (X.W)^T (X.W) - (X.W)^T.Y - Y^T (X.W)$$

On utilise la propriété $(AB)^T = B^T A^T$

$$= \nabla_w W^T X^T X.W - (X.W)^T.Y - Y^T(X.W)$$

On utilise le fait que $X^T X$ soit une matrice symetrique et $\nabla_x x^T Ax = 2Ax$,

$$= 2X^T XW + \nabla_w - (X.W)^T.Y - Y^T X.W$$

On utilise la propriéte $\nabla A^T = (\nabla A)^T$

$$= 2(X^T XW - X^T Y)$$

On souhaite maintenant annuler le gradient, donc on pose :

$$\nabla_w MSE(W) = 0$$

$$2(X^T XW - X^T Y) = 0$$

$$X^T XW = X^T Y$$

On suppose $X^T X$ inversible, donc on a :

$$W = (X^T X)^{-1} X^T Y$$

Ne peut pas être calculé en général, car $X^T X$ rarement inversible (les dimensions de X sont souvent corrélées). Pour ce faire il faut perturber la matrice $X^T X$ avec un certain λ tel que (I la matrice identité):

$$W = (X^T X + \lambda I)^{-1} X^T Y$$

Cette astuce se traduit dans la MSE par un terme de généralisation, tel que :

$$MSE(W) = (X.W - Y)^T (X.W - Y) + \lambda ||W||$$

λ est un hyper paramètre que l'on peut fixer pour régler les problèmes de sur et sous apprentissages dans les modèles linéaires.

4.2.4 Q2.1.4

On définit le biais qui correspond à $w_0 \in \mathbb{R}$, que l'on intègre à la fonction MSE(W). Ce qui donne sous forme indicielle :

$$MSE(W) = \sum_{i=1}^{n} (f_w(x_i) - y_i)^2 = \sum_{i=1}^{n} ((w_1 x_{i,1} + ... + w_d x_{i,d} + w_0) - y_i)^2$$

Sous forme matricielle :

$$MSE(W) = (X.W + w_0 - Y)^T (X.W + w_0 - Y)$$

On cherche donc, comme avant à annuler la dériver pour obtenir le w_0^*, ce qui donne :

$$\frac{dMSE(W)}{dw_0} = 0$$

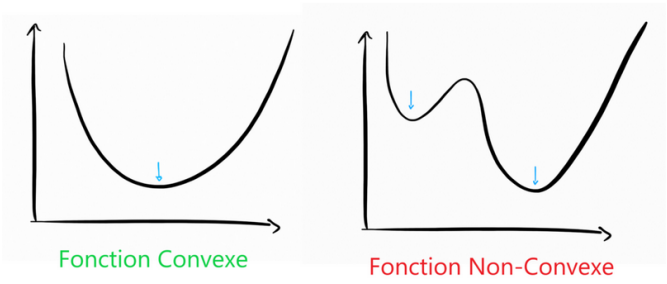
$$\frac{d}{dw_0} \frac{1}{N} \sum_{i=1}^{N} (f_w(x_i) + w_0 - y_i)^2 = 0$$

$$\frac{1}{N} \sum_{i=1}^{N} 2(f_w(x_i) + w_0 - y_i) = 0$$

$$w_0^* = \frac{2}{N} (\sum_{i=1}^{N} y_i - \sum_{i=1}^{N} f_w(x_i))$$

4.2.5 Q2.2

Définition Descente de gradient : La Descente de Gradient est un algorithme d'optimisation qui permet de trouver le minimum de n'importe quelle fonction convexe en convergeant progressivement vers celui-ci.



Algorithme de descente de gradient dans le cadre de la régression linéaire :

Version BATCH (on utilise toutes les données de X à chaque itération) :

- On initialise aléatoirement W, les paramètres du modèle.
- Pour i allant de 1 à $IterationMax$ ou convergence faire :

$$W = W - \alpha * \frac{1}{N} \nabla_w MSE(W) \text{ (la moyenne)}$$
$$W = W - \alpha \frac{1}{N} 2(X^T XW - X^T Y)$$

Version STOCHASTIQUE (on utilise une donnée à chaque itération) :

- On initialise aléatoirement W, les paramètres du modèle.
- On shuffle les Data : shuffle(data)
- Pour i allant de 1 à N ou convergence faire :

$$W = W - \alpha(x^{i^T} x^i W - x^{i^T} y^i)$$

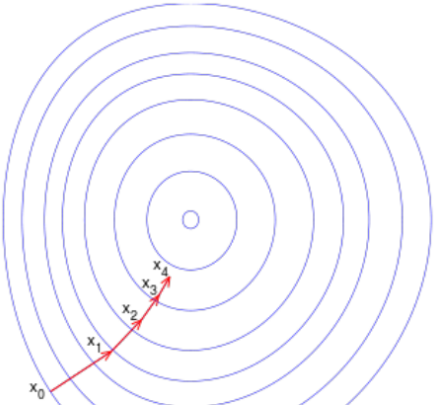
Version MINI-BATCH (on utilise une partition des données à chaque itération) :

- On initialise aléatoirement W, les paramètres du modèle.
- Pour i allant de 1 à $NbPartition$ ou convergence faire :

$$W = W - \alpha(X^{i^T} X^i W - X^{i^T} y^i)$$

4.2.6 Q2.3.1

Si on prends un α très grand, alors on risque de diverger en faisant de grand pas. C'est à dire que l'on naviguera autour du minimum sans jamais l'atteindre. Au contraire, avec un α très petit le temps de convergence jusqu'au minimum sera très lent.

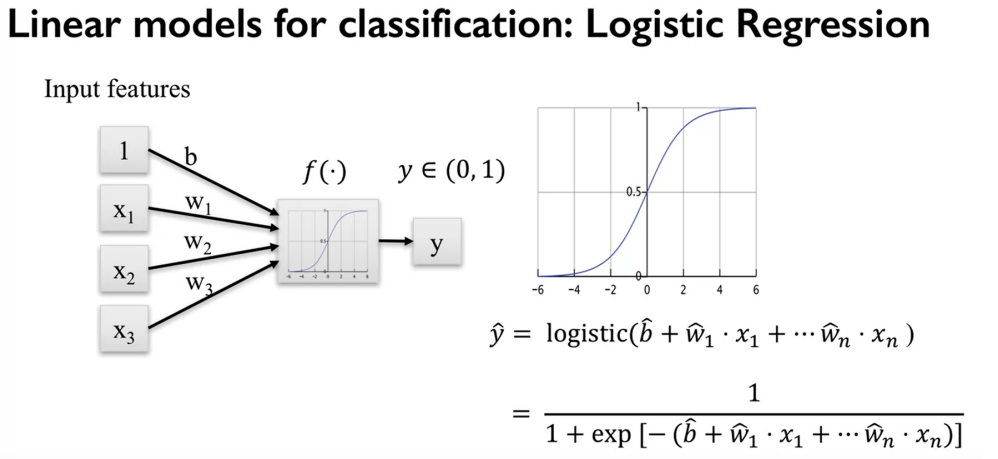


4.2.7 Q2.3.2

TODO

4.3 Exercice 3 – Régression logistique

Définition Régression logistique : La régression logistique est un modèle statistique permettant d'étudier les relations entre un ensemble de variables qualitatives Xi et une variable qualitative Y. Il s'agit d'un modèle linéaire généralisé utilisant une fonction logistique comme fonction de lien.
Un modèle de régression logistique permet aussi de prédire la probabilité qu'un événement arrive (valeur de 1) ou non (valeur de 0) à partir de l'optimisation des coefficients de régression. Ce résultat varie toujours entre 0 et 1. Lorsque la valeur prédite est supérieure à un seuil, l'événement est susceptible de se produire, alors que lorsque cette valeur est inférieure au même seuil, il ne l'est pas.



La régression logistique utilise une forme spécial pour le modèle prédictif. Il définit la probabilité de la première classe comme :

$$P(C_1|x) = \sigma(w^T x + b)$$

Comme il s'agit d'un paramètre de classification binaire, la probabilité d'appartenir à l'autre classe est:

$$P(C_2|x) = 1 - P(C_1|x)$$

Ainsi, l'étiquette de classe prédite y est la classe avec la probabilité la plus élevée.
La fonction sigmoïde est définie comme suit :

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$x = log(\frac{\sigma(x)}{1-\sigma(x)})$$

Notez que le modèle prédictif utilise la fonction sigmoïde pour calculer la probabilité. Ceci est possible parce que la sortie de la fonction sigmoïde est limitée dans la plage [0,1].

4.3.1 Q3.1

La forme générale du classifieur binaire f.

$$f_w(x) = \begin{cases} 1 & \text{si } \sigma(w^T x + b) \geq 0.5 \\ -1 & \text{sinon.} \end{cases} \quad = argmax_{y \in \{y_1, y_2\}} P(y = y_i|x)$$

Pour connaître la probabilité associé d'obtenir la classe y_{pred} en fonction d'un exemple x :

$$P(y_{pred}|x) = max_{y\in\{y_1,y_2\}} P(y = y_i|x)$$

- Quel est le but ? : Classifier des données binairement.
- Quelle étiquette prédire pour x si $w.x > 0$?

Si $w.x > 0$ alors on a :

$$\frac{P(y=1|x)}{1-P(y=1|x)} > 1$$

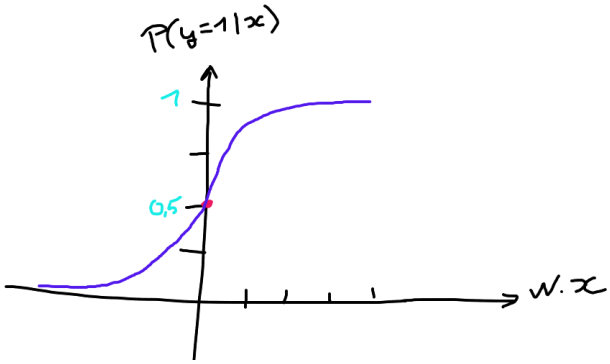
Donc

$$P(y = 1|x) > P(y \neq 1|x)$$

Donc il est plus probable que la classe 1 soit bonne lorsque $w.x > 0$. On obtient le classifieur associé :

$$f_w(x) = \begin{cases} 1 & \text{si } w.x > 0 \\ -1 & \text{sinon.} \end{cases}$$

- Que vaut p(y = 1|x) ? Tracer la fonction p(y = 1|x) en fonction de w.x.



$P(y = 1|x) = \frac{1}{1+e^{-wx}}$ (sigmoide)

Nous savons que lorsque $w.x > 0$, nous avons plus de chance de prédire la classe 1, donc que sa probabilité soit > 0.5 La courbe est majoré par 0 et 1 car $P(y = 1|x)$ représente la probabilité que l'on prédise 1.

De manière générale :

$$\begin{aligned} wx + b < 0 &\implies \sigma(wx + b) < 0.5 \implies P(y = 1|x) < 0.5 \\ wx + b = 0 &\implies \sigma(wx + b) = 0.5 \implies P(y = 1|x) = 0.5 \\ wx + b > 0 &\implies \sigma(wx + b) > 0.5 \implies P(y = 1|x) > 0.5 \end{aligned}$$

- Type de frontière de la régression logistique
C'est une frontière linéaire car notre décision se base sur les coefficients $w^T x + b$ pour prendre une décision. Ce calcul étant linéaire alors notre frontière de décision sera linéaire.

$$\begin{aligned} P(y = -1|x) &= 1 - P(y = 1|x) \\ &= 1 - \frac{1}{1+e^{-wx}} \\ &= \frac{1+e^{-wx}-1}{1+e^{-wx}} \\ &= \frac{e^{-wx}}{1+e^{wx}} \\ &= \frac{1}{1+e^{wx}} \end{aligned}$$

Plus généralement, on a :

$$\begin{aligned} P(y = y_i|x) &= \frac{1}{1+e^{-y_i wx}} \\ &= \sigma(y_i wx) \end{aligned}$$

4.3.2 Q3.2

Si le poids est nul alors pas d'influence de x.
Si le poids est positif alors plus x_i augmente plus $P(y = 1|x_i)$ augmente.

Limite : il faut que les données soit linéairement séparable.

4.3.3 Q3.3

La vraisemblance de w par rapport a un exemple (x,y) correspond à:

$$L = P(y|x) = \prod_{i=1}^{n} P(y_i|x) = \prod_{i=1}^{n} \frac{1}{1+e^{-ywx}}$$

La log vraisemblance :

$$\begin{aligned} LL &= log \prod_{i=1}^{n} \frac{1}{1+e^{-ywx}} \\ LL &= \sum_{i=1}^{n} log \frac{1}{1+e^{-ywx}} \\ LL &= -\sum_{i=1}^{n} log(1 + e^{-ywx}) \end{aligned}$$

4.3.4 Q3.4

Pour résoudre le problème on fait une montée de gradient car on utilise la maximisation de la log vraisemblance.
On calcul dont le gradient :

$$\begin{aligned} \frac{dLL}{dw_j} &= \frac{dLL}{dw_j}(-\sum_{i=1}^{n} log(1 + e^{-ywx}) \\ \frac{dLL}{dw_j} &= -\sum_{i=1}^{n} \frac{\frac{d}{dw_j}(-y_i(xw))e^{-y_i wx}}{1+e^{-ywx}} \\ \frac{dLL}{dw_j} &= -\sum_{i=1}^{n} y^i x_j^i * \frac{1}{1+e^{y^i(x^i w)}} \end{aligned}$$

Montée de gradient :

$$w = w + \epsilon \frac{dLL}{dw_j}$$

5 TD 4 : Perceptron

5.1 Exercice 1 – Perceptron

Le perceptron est l'unité de base des réseaux de neurones. Il s'agit d'un modéle de classification binaire, capable de séparer linéairement 2 classes de points.

5.1.1 Q1.1

$y \in \{-1, 1\}$
La fonction cout au sens des moindres carres:

$$Mean\_Square\_Error(f_w(x), y) = \frac{1}{N}\sum(\hat{y} - y)^2 = (f_w(x) - y)^2 = ||f_w(x) - y||^2$$

Avec
$f_w(x) =< x, w > +b = \sum_i (w_i x_i) + b$
$\hat{y}$ est la prédiction du modéle, la classe choisis ,
y est la vrai classe.

5.1.2 Q1.2

Avec l'algorithme du perceptron on utilise comme fonction coût :

$$L(f_w(x), y) = max(0, -yf_w(x))$$

SI :

$y+, f_w(x)+$
$y-, f_w(x)-$
Alors L = 0

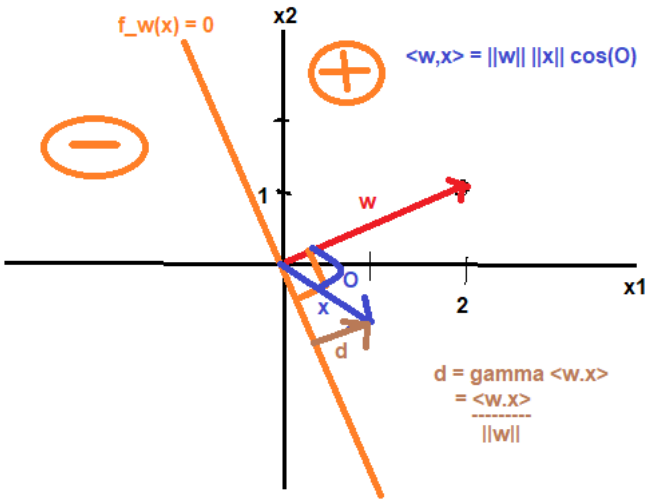
SI :

$y+, f_w(x)-$
$y-, f_w(x)+$
Alors L $>$ 0

5.1.3 Q1.3

Étant donné qu'on a le droit a une fonction f de complexité infinie, (on peut faire ce qu'on veut) il nous suffit d'entourer les cercles rouges. Cependant, cela mène a de l'overfitting donc pas intéressant.

5.1.4 Q1.4



Le vecteur W pointe toujours vers les valeurs qui augmentent.
Pour un point mal classé disons, x = (-1,-1) avec comme classe y = 1 :
On fait la mise a jour :

$$w^{'} = w + yx$$
$$w^{'} = \begin{pmatrix}2\\1\end{pmatrix} + 1 * \begin{pmatrix}-1\\-1\end{pmatrix}$$
$$w^{'} = \begin{pmatrix}1\\0\end{pmatrix}$$

avec un $w^{'}$ positif donc toujours mal classé, on continue la mise a jour :

$$w^{''} = w^{'} + yx$$
$$w^{''} = \begin{pmatrix}1\\0\end{pmatrix} + 1 * \begin{pmatrix}-1\\-1\end{pmatrix}$$
$$w^{''} = \begin{pmatrix}0\\-1\end{pmatrix}$$

Et la le point est bien classé. Pour le produit scalaire :

$$w.x = -3$$
$$w^{'}.x = -1$$
$$w^{''}.x = +1$$

5.1.5 Q1.5

w^1, w^2 et w sont colinéaires / proportionnels donc la frontière de décision est la même.
Alors que pour w^3 la frontière de décision est inversé, on inverse les classes.

5.1.6 Q1.6

Pour montrer, on fait le gradient de la Loss :

$$w^{t+1} = w^t - \alpha \nabla_w L(f_w(x), y)$$
$$w^{t+1} = w^t - \alpha \nabla_w max(0, -y f_w(x)$$

SI :
$-yf_w(x) < 0$ Alors 0 Et on ne fait rien

Sinon :
$-y(< w.x > +b) > 0$ Alors $-yx$ et on a donc :

$$w^{t+1} = w^t + \alpha yx$$

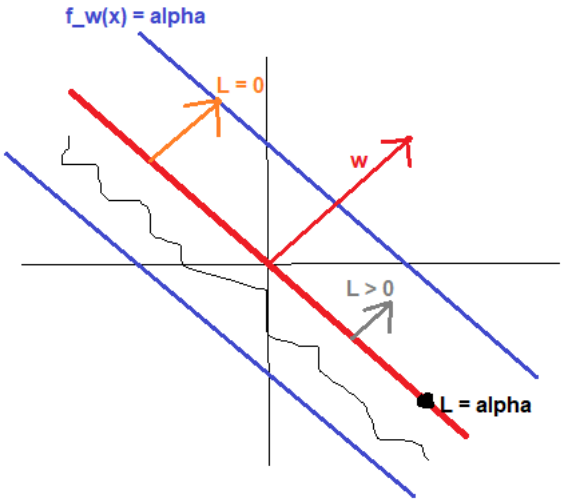
Qui correspond a la descente de gradient.

5.1.7 Q1.7

On a des problèmes quand $w = 0$ car le perceptron ne bouge pas. Pour regler les problemes, on modifie la fonction de coût en introduisant une marge :

$$max(0, \alpha - yf_w(x)$$

SI : $yf_w(x) > \alpha$ alors 0
Sinon : $-yx$



5.1.8 Q1.8

Version BATCH (On fait une descente de gradient sur D) :

- On initialise aléatoirement W, les paramètres du modèle.
- Pour i allant de 1 à $IterationMax$ ou convergence faire :

$$W = W - \alpha * \sum_1^N \frac{1}{N} \nabla L(f(x^i), y^i) \text{ (la moyenne)}$$

Version STOCHASTIQUE (échantillon de taille 1) / perceptron :

- On initialise aléatoirement W, les paramètres du modèle.
- On shuffle les Data : shuffle(data)
- Pour i allant de 1 à N ou convergence faire :

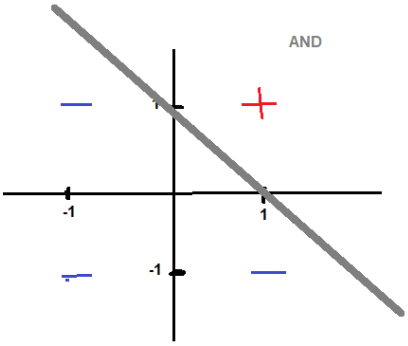
$$W = W - \alpha \nabla L(f(x^i), y^i)$$

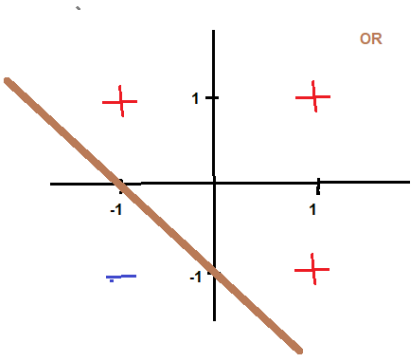
Version MINI-BATCH (sur un sous- ensemble de D) :

- On initialise aléatoirement W, les paramètres du modèle. $(X_1, ..., X_D)$ avec $\cap X_i = \emptyset$ et $\cup X_i = X$
- for X_i :

$$W = W - \alpha \nabla L(f(X^i), y^i)$$

5.1.9 Q1.9





Pour and : $w = (1, 1, 1)$ et donc $f_w(x) = 1 + x_1 + x_2$
Pour or : $w = (-1, 1, 1)$ et donc $f_w(x) = -1 + x_1 + x_2$

5.2 Exercice 2 – Convergence du Perceptron

5.2.1 Q2.1

γ comme vu dans l'exercice 1 correspond a la marge pour résoudre le problème du zéro. w^* n'est pas unique comme vu aussi dans l'exercice 1 lorsqu'on prend des w colinéaires.
On va maintenant démontrer le théorème de Novikoff :

$$t^* \leq \frac{R^2}{\gamma^2}$$

avec t^* le nombre maximum d'itérations avant convergene.

5.3 Exercice 3 – Expressivité des séparateurs linéaires

On se place dans l'espace des séparateurs linéaires : $f_w(x) = \sum_{i=1}^{N} x_i w_i$

5.3.1 Q3.1

La dimension de w est d.
L'écriture matricielle est : $f_w(x) =< x.w >$

5.3.2 Q3.2

- La dimension de w est 6 car 6 éléments.

- La projection correspond a une fonction quadratique, a la frontière de décision:

$$f_w(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x_1 x_2$$

Si :
$w_4 w_5 = 0$ alors on aura une parabole

$w_4 w_5 > 0$ alors on aura une ellipse

$w_4 w_5 < 0$ alors on aura une hyperbole

Le modèle linéaire est un cas particulier $w_0 = ... = w_5 = 0$

5.3.3 Q3.3

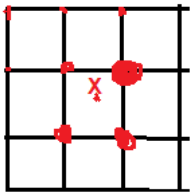
- La frontière est plus intéressante avec la nouvelle représentation.

- Le coût sera inférieur (car le modèle linéaire est un cas particulier)

- On se base sur le taux de bonne classification (sur des données de 'validations')

5.3.4 Q3.4

On fait une grille de taille N^2



On mesure la similarité gaussienne du point c par rapport à chaque point de la grille :

$$s(x, p^{i,j}) = Ke^{-\frac{||x-p^{i,j}||^2}{\sigma}}$$

- w a comme dimension d,

- Laformule de la frontière de décision :

$$f_w(x) = \sum_{i,j=1}^{n} w_{i,j} s(x, p^{i,j})$$

6 TD 5 : SVM

6.1 Exercice 1 – Support Vector Machine

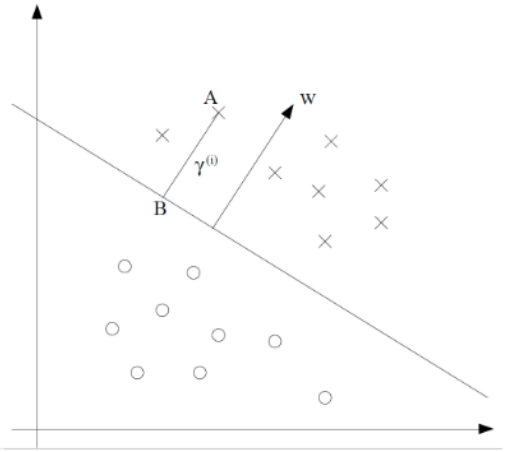


FIGURE 1 – Ensemble de données linéairement séparables

6.1.1 Q1.1.1

A quoi est égal γ^i ?
On remarque que $\vec{BA} = y^i * \gamma^i \frac{w}{||w||}$ (le projeté de x^i sur l'hyperplan)
On sait que

$\frac{w}{||w||}$: vecteur unitaire orthogonale à la frontière de décision f.

On pose alors :

$$\vec{BA} = y^i * \gamma^i \frac{w}{||w||}$$

$$x^i - x_B = y^i * \gamma^i \frac{w}{||w||}$$

$$x_B = x^i - y^i * \gamma^i \frac{w}{||w||}$$

On utilise le fait que x_B appartienne à l'hyperplan, c'est à dire :

$$x_B.w + b = 0$$

Ce qui donne

$$(x^i - y^i * \gamma^i \frac{w}{||w||}).w + b = 0$$

$$x^i.w - y^i * \gamma^i \frac{||w||^2}{||w||} + b = 0$$

$$x^i.w - y^i * \gamma^i ||w|| + b = 0$$

$\gamma^i = y^i \frac{x^i.w+b}{||w||}$ car $y^i \in \{-1, 1\}$ alors $\frac{1}{y^i} = y^i$

6.1.2 Q1.1.2

Comment faire pour que min γ^i soit maximisé ?

On remarque que la solution $(\alpha w, \alpha b)$ ne change pas la distance γ^i car

$$\gamma_2^i = y^i \frac{\alpha(x^i.w)+\alpha b}{\alpha||w||} = y^i \frac{(x^i.w)+b}{||w||} = \gamma^i$$

On cherche la solution en posant la contrainte :

$$min_i y^i(w.x^i + b) = 1$$

Dans ce cas la,

$$min_i \gamma^i = \frac{1}{||w||}$$

Donc maximiser la marge $min_i \gamma^i$ est équivalent à maximiser $\frac{1}{||w||}$ donc minimiser $||w||$.

6.1.3 Q1.2

On considère alors le problème d'optimisation sous contraintes suivant :

$$min \frac{1}{2}||w||^2 \text{ tel que }, y^i(x^i.w + b) \geq 1$$

6.1.4 Q1.2.1

Pourquoi choisit-on la contrainte ≥ 0 **?**
Dans ce cas, il y a une solution triviale $w^* = 0$ et $b^* = 0$. Mais ce n'est pas satisfaisant car tous les points appartiennent à la frontière de décision.
Pourquoi choisit-on la contrainte ≥ 1 **?**
Il n'y a pas de raison spéciale pour 1, n'importe quel nombre de $\mathbb{R}_+^*$ fait l'affaire.

6.1.5 Q1.2.2

Posons le lagrangien :

$$L(w, b, a) = \frac{1}{2}||w||^2 + \sum_{i=1}^{N} a_i(1 - y^i(x^i.w + b)) \text{ avec } a_i \geq 0$$

Le problème devient alors :

$$min_{w,b} max_a L(w, b, a) \text{ avec } a_i \geq 0$$

Si $1 - y^i(x^i.w + b) > 0$ alors $y^i(x^i.w + b) < 1$ donc la contrainte n'est pas vérifiée.

Ce qui donne $max_a L = +\infty$, alors la solution est non retenu si le problème est linéairement séparable.

6.1.6 Q1.2.3

Solution analytique du lagrangien :

Propriété : $min_x max_y f(x, y) = max_y min_x f(x, y)$ ssi f est convexe en x et concave en y.

On peut appliquer la propriété car c'est notre cas, le problème devient alors :

$$max_a min_{w,b} L(w, b, a)$$

Si on obtient une solution analytique avec la minimisation $min_{w,b} L(w, b, a)$, alors on a gagné. Nous allons donc annulé le gradient de L par rapport à w et b.

Par rapport à w :

$$\nabla_w L = 0$$

$$\nabla_w(\frac{1}{2}||w||^2 + \sum_{i=1}^N a_i(1 - y^i(x^i.w + b))) = 0$$

$$w - \sum_{i=1}^N a_i y^i x^i = 0$$

$$w = \sum_{i=1}^N a_i y^i x^i$$

Par rapport à b :

$$\frac{dL}{db} = 0$$

$$-\sum_{i=1}^N a_i y^i = 0$$

On a donc une nouvelle contrainte :

$$\sum_{i=1}^N a_i y^i x^i = 0$$

6.1.7 Q1.2.4

Formulation duale de notre problème d'optimisation :

On veut maximiser $L(w, b, a)$ par rapport à a en utilisant w et b de **1.2.3**.

$$max_a \frac{1}{2}||w||^2 + \sum_{i=1}^N a_i(1 - y^i(x^i.w + b))$$

$$= max_a \frac{1}{2}||\sum_{i=1}^N a_i y^i x^i||^2 - b\sum_{i=1}^N a_i y^i + \sum_{i=1}^N a_i - \sum_{i=1}^N a_i y^i(\sum_{j=1}^N a_j y^j x^j.x^i)$$

$$= max_a \sum_{i=1}^N a_i + \frac{1}{2} < \sum_{i=1}^N a_i y^i x^i, \sum_{j=1}^N a_j y^j x^j > - \sum_{i=1}^N \sum_{j=1}^N a_i a_j y^i y^j x^i x^j$$

$$= max_a \sum_{i=1}^N a_i - \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N a_i a_j y^i y^j x^i x^j$$
$$\text{avec } a_i \geq 0 \text{ et } \sum_{i=1}^N a_i y^i = 0$$

6.1.8 Q1.2.5

Le problème devient :

$$max_a \sum_{i=1}^N a_i - \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N a_i a_j y^i y^j x^i x^j$$

On voit que l'on à juste à verifier le produit scalaire $x^i.x^j$, donc on peut utiliser à la place n'importe quelle noyau tel que $K(x^i, x^j) =< \phi(x^i), \phi(x^j) >$

6.1.9 Q1.2.6

Si le problème n'est pas linéairement séparable, alors il n'existe pas de solution. On utilise alors des marges souples, on introduit une pénalisation pour obtenir :

$$min_{w,b,\psi_i} \frac{1}{2}||w||^2 + C\sum_{i=1}^N \psi_i \text{ ou } \forall i, \psi_i \geq 0 \text{ et } y^i(x^i.w + b) \geq 1 - \psi_i$$

6.1.10 Q1.2.7

Comme la réecriture du problème de **1.2.2** à **1.2.4** a introduit un lagrangien,

$$L = \frac{1}{2}||w||^2 + C\sum_{i=1}^N \psi_i + \sum_{i=1}^N a_i(1 - \psi_i - y^i(x^i.w + b)) - \sum_{i=1}^N \beta_i \psi_i$$

Ce qu'on veut, c'est :

$$min_{w,b,\psi} max_{a,\beta} L$$

On passe donc au Dual avec la propriété min max = max min.

On pose donc :

$$\frac{dL}{d\psi_i} = 0$$

$$C - a_i - \beta_i = 0$$

$$a_i + \beta_i = C$$

On sait que $\beta_i \geq 0$ donc $a_i \leq C$

En multipliant, on obtient le problème :

$max_a \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j y^i y^j x^i x^j$ avec $0 \leq a_i \leq C$ et $\sum_{i=1}^{N} a_i y^i = 0$

6.1.11 Q1.2.8

La fonction utilisée par la classification est :

$$f(x, w, b) = w.x + b$$

$$= \sum_{i=1}^{N} a_i y^i x^i.x + b \text{ (avec } \mathbf{1.2.3}\text{)}$$

$$= \sum_{i=1}^{N} a_i y^i K(x^i, x) + b$$

6.1.12 Q1.2.9

KKT : $a_i(1 - \psi_i - y^i(w.x + b)) = 0$

On a plusieurs cas :

$$a_i = C \implies \beta_i = 0 \implies \psi_i \geq 0 \implies \text{point } x^i \text{ est mal classifié.}$$

$$a_i = 0 \implies \beta_i = C \implies \psi_i = 0 \implies \text{point } x^i \text{ est bien classifié.}$$

$$0 < a_i < C \implies 0 < \beta_i < C \implies \text{point } x^i \text{ est bien classifié.}$$

En utilisant la condition KKT, c'est à dire :

$$a_i(1 - \psi_i - y_i(w.x + b)) = 0 \implies y^i(w.x + b) = 1$$

6.1.13 Q1.2.10

On prend un point x^i tel que $0 < a_i < C$

$$\implies b = 1 - y^i w.x^i$$

6.2 Exercice 2 - Noyaux

K est un noyau ssi $\exists \phi$ tel que $K(x, y) =< \phi(x), \phi(y) >$

6.2.1 Q2.1

$C \geq 0$

$$CK(x,y) = C < \phi(x), \phi(y) > \text{ car } K \text{ est un noyau.}$$

Sachant que $< ., . >$ est un produit scalaire, on sait que :

$$< ax, y >= a < x, y >$$

$$< x, ay >= a < x, y >$$

donc :

$$=< \sqrt{C}\phi(x), \sqrt{C}\phi(y) >$$

$\phi'(x) = \sqrt{C}\phi(x)$ est la fonction de projection associée à $CK \implies CK$ est un noyau.

6.2.2 Q2.2

$$K(x,y) + K'(x,y)$$

$$=< \phi(x), \phi(y) > + < \phi'(x), \phi'(y) >$$

$$\sum_{i=1}^{N} \phi_i(x)\phi_i(y) + \sum_{j=1}^{N} \phi'_j(x)\phi'_j(y)$$

$$=< concat(\phi(x), \phi'(x)), concat(\phi(y), \phi'(y)) >$$

6.2.3 Q2.3

$$KK'$$

$$= K(x,y)K'(x,y)$$

$$=< \phi(x), \phi(y) >< \phi'(x), \phi'(y) >$$

$$= \sum_{i=1}^{N} \phi_i(x)\phi_i(y) * \sum_{j=1}^{N} \phi'_j(x)\phi'_j(y)$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} \phi_i(x)\phi_i(y) * \phi'_j(x)\phi'_j(y)$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} \phi_i(x)\phi'_j(x)\phi_i(y)\phi'_j(y)$$

On pose : $\phi''_{i,j}(x) = \phi_i(x)\phi'_j(x)$

$$=< \phi''(x), \phi''(y) >$$